Predictive Maintenance using Python (Google Colab)

Table of Contents

Course Objective

To equip learners with the ability to perform predictive maintenance using Python-based tools and libraries. This course walks through creating a sample dataset, analyzing it visually, building a machine learning model, and making predictions to prevent machinery failure.

Setup and Introduction

Environment Setup

Ensure you're using **Google Colab** or any Python 3.x environment with the following libraries:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report,
confusion_matrix
```

Learning Outcomes

- Understand what predictive maintenance is
- Create and visualize machine data
- Train a machine learning model
- Evaluate the model and predict failures

Creating Sample Dataset

```
data = {
    'MachineID': range(1, 11),
    'Temperature': [65, 70, 75, 80, 85, 90, 95, 100, 105, 110],
    'Vibration': [0.02, 0.03, 0.04, 0.04, 0.05, 0.06, 0.08,
0.09, 0.1, 0.11],
    'WorkingHours': [1000, 1200, 1500, 1600, 1800, 2000, 2200,
2400, 2700, 3000],
    'Failure': [0, 0, 0, 0, 1, 1, 1, 1, 1, 1]
}
df = pd.DataFrame(data)
df.head()
```

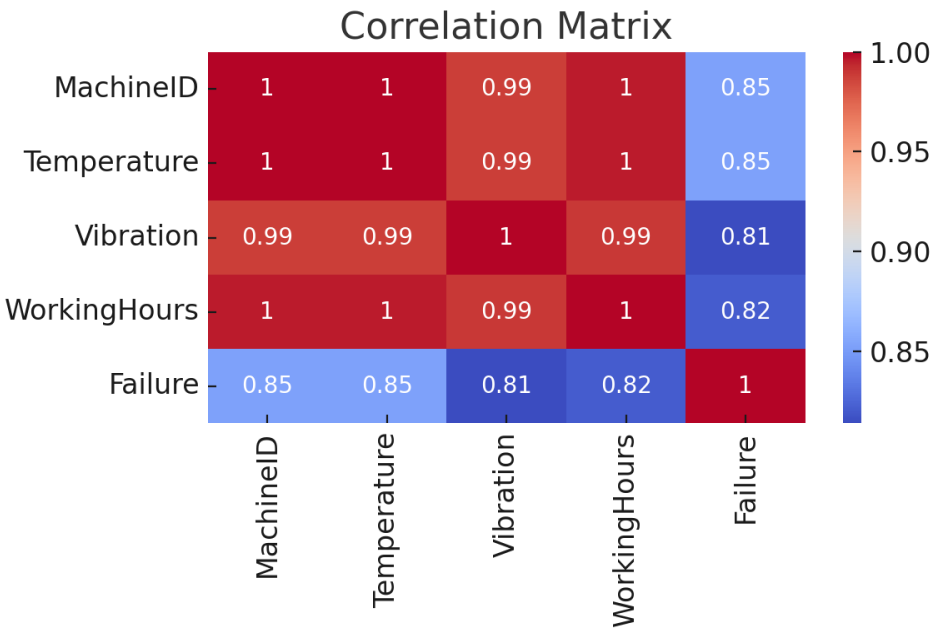
Sample Dataset Preview

MachineID	Temperature	Vibration	WorkingHours	Failure
1	65	0.02	1000	0
2	70	0.03	1200	0
3	75	0.04	1500	0
4	80	0.04	1600	0
5	85	0.05	1800	1

Data Visualization

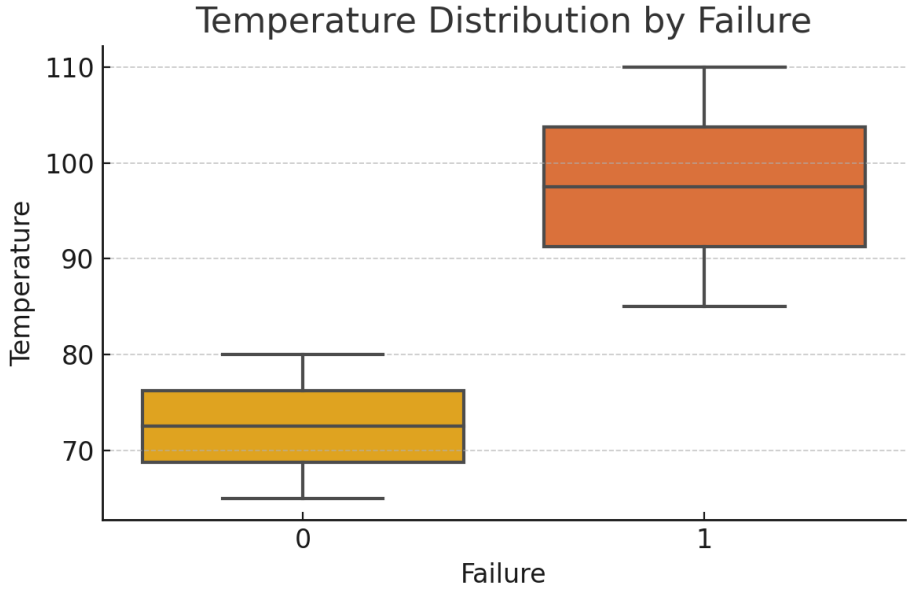
Correlation Matrix

```
plt.figure(figsize=(6, 4))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Matrix")
plt.tight_layout()
plt.show()
```



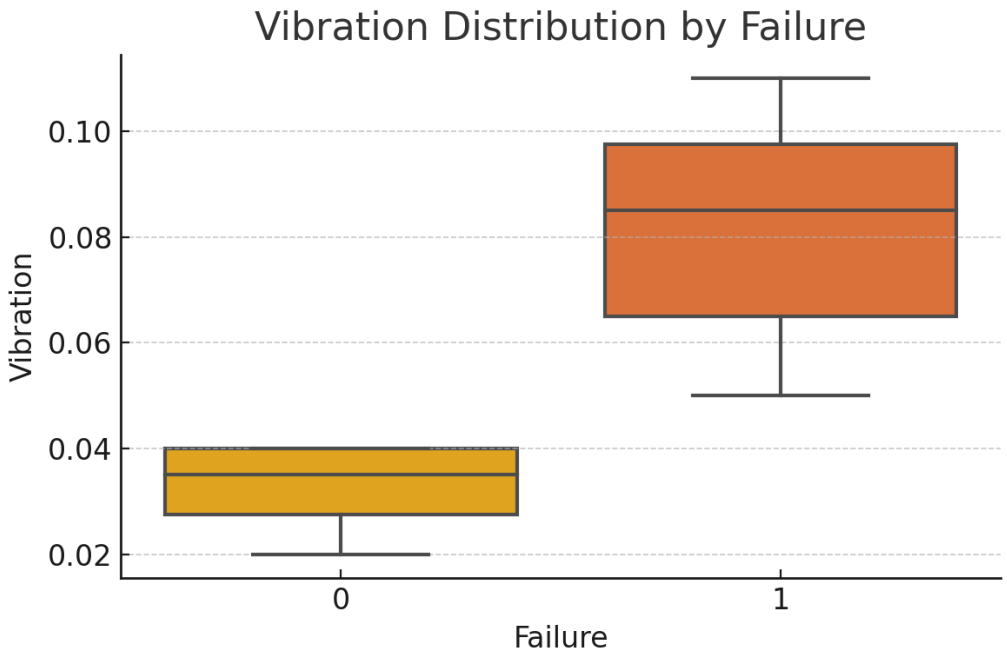
Temperature vs Failure

```
plt.figure(figsize=(6, 4))
sns.boxplot(x='Failure', y='Temperature', data=df)
plt.title("Temperature Distribution by Failure")
plt.tight_layout()
plt.show()
```



Vibration vs Failure

```
plt.figure(figsize=(6, 4))
sns.boxplot(x='Failure', y='Vibration', data=df)
plt.title("Vibration Distribution by Failure")
plt.tight_layout()
plt.show()
```



Model Building

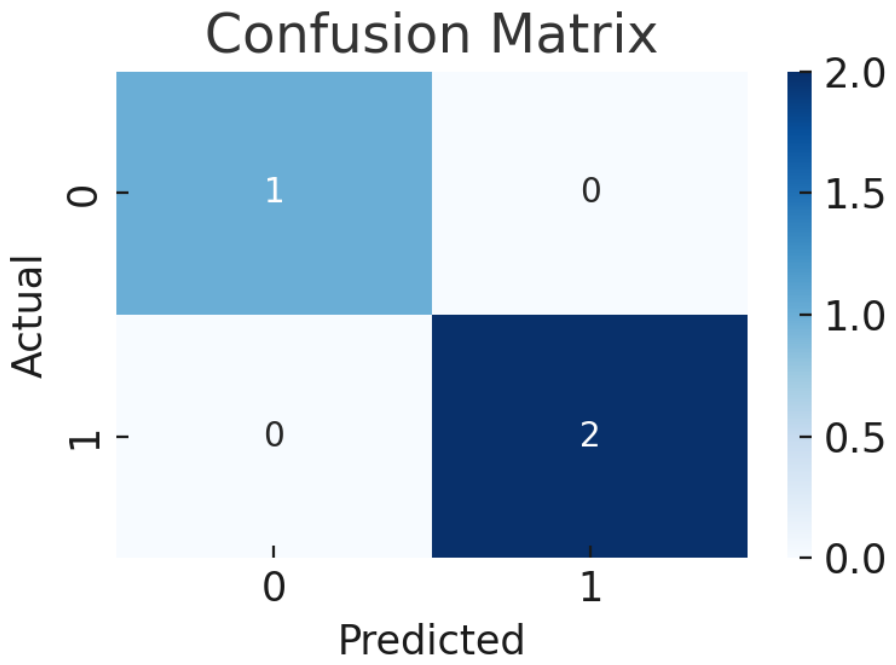
Training the Model

```
X = df[['Temperature', 'Vibration', 'WorkingHours']]
y = df['Failure']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

model = RandomForestClassifier(n_estimators=100,
random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Model Evaluation

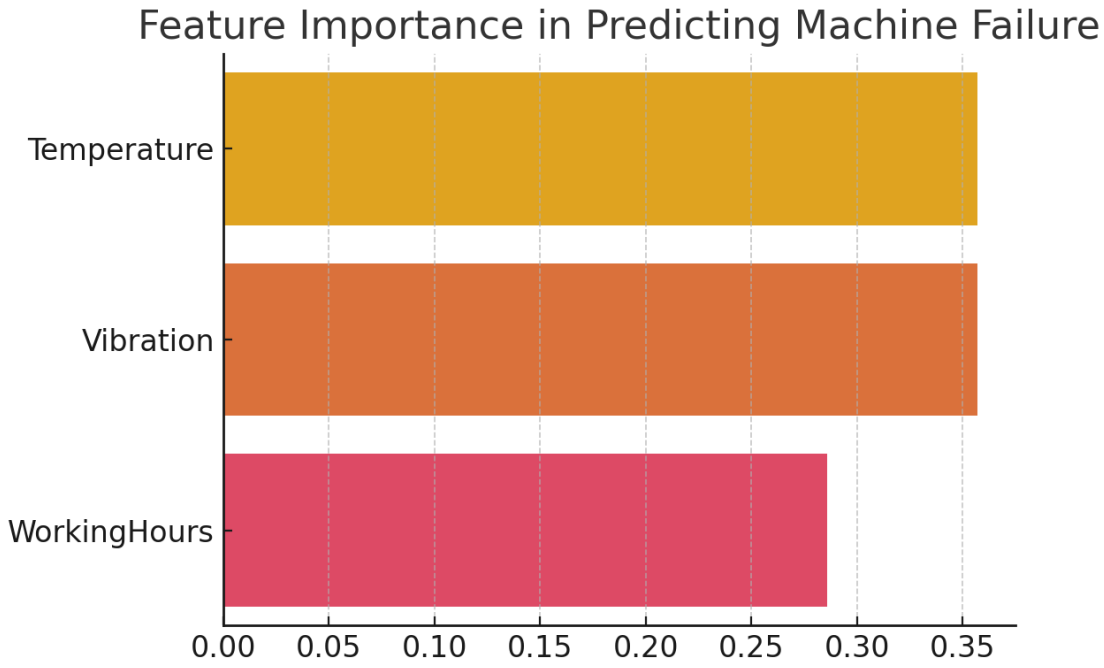
Classification Report and Confusion Matrix

```
print("Classification Report:\n", classification_report(y_test,
y_pred))
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(4, 3))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.tight_layout()
plt.show()
```



Feature Importance

```
importances = model.feature_importances_
features = X.columns

plt.figure(figsize=(6, 4))
sns.barplot(x=importances, y=features)
plt.title("Feature Importance in Predicting Machine Failure")
plt.tight_layout()
plt.show()
```



Prediction for New Data

```
new_machine = pd.DataFrame({
    'Temperature': [98],
    'Vibration': [0.08],
    'WorkingHours': [2300]
})
prediction = model.predict(new_machine)
print("Predicted Failure (1=Yes, 0=No):", prediction[0])
```

Future Extensions

- Integrate real-world sensor datasets (e.g., NASA Turbofan Engine Dataset)
- Deploy models via Streamlit for web visualization
- Include time-series analysis for early failure detection

Github Repository

https://github.com/deepakrll/ML-Mechanical-Preventive-Maintenance